

UNIVERSITY OF CANTERBURY

HONOURS PROJECT

**Identifying and Responding to  
Disengagement in a  
Constraint-Based Intelligent  
Tutoring System**

*Author:*

Jin Kwang HONG

*Supervisors:*

Dr. Antonija MITROVIC

Dr. Kourosh NESHATIAN

October 21, 2014

## **Abstract**

Numerous studies have shown the effectiveness of Intelligent Tutoring Systems in improving learning. However, there are still students who do not engage with problem solving and therefore miss opportunities for learning. In my project, I am investigating whether it is possible to predict when the student would abandon a problem, thus allowing the system to try to motivate the student to persevere. I will discuss the approach taken to generate a predictor, including identifying the features from log data and the selection of the learning algorithm to produce a predictor. To demonstrate the accuracy of the generated decision tree predictor, we conduct an experiment deploying this predictor in a study with SQL-Tutor.

# Contents

1	Introduction . . . . .	3
1.1	Intelligent Tutoring Systems . . . . .	3
1.2	Motivation . . . . .	4
1.3	Goals . . . . .	5
1.4	Report Structure . . . . .	5
2	Background . . . . .	5
2.1	SQL-Tutor . . . . .	6
2.2	Modelling Off-task Behaviour . . . . .	8
2.3	Gaming in Constraint-Based Tutors . . . . .	9
2.4	Disengagement Detection using Mouse Movement In-formation . . . . .	9
2.5	Disengagement Detection using Eye Tracking . . . . .	10
2.6	Detecting Quitting Behaviour in Reading . . . . .	10
3	Predictor . . . . .	11
3.1	Datasets . . . . .	14
3.2	Features . . . . .	15
3.3	Choosing the Best Predictive Model . . . . .	19
4	Experiment . . . . .	25
4.1	Goals and Hypotheses . . . . .	25
4.2	Intervention Design . . . . .	25
4.3	Procedure . . . . .	27
4.4	Participants and Apparatus . . . . .	29
5	Results . . . . .	30
5.1	Predictor Accuracy . . . . .	32
5.2	Effectiveness of Intervention . . . . .	34
6	Discussion . . . . .	37
7	Conclusion . . . . .	38

	7.1	Limitations . . . . .	39
	7.2	Future Work . . . . .	39
8		Appendix . . . . .	42

# 1 Introduction

## 1.1 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are computer systems that aim to emulate one on one human to human tutoring [16]. ITS have been developed for a number of different domains such as SQL (SQL-Tutor) [12], or Linked List Structures (iList) [8] and have shown to have a positive effect on learning [13]. ITS analyse the submissions made by students and also use the student's historical data to provide meaningful feedback on the correctness of the solutions. The historical data of students is referred to as the *student model* [16].

There are two student modelling approaches that are common in today's ITS: model tracing and constraint-based modelling. Model tracing involves following students' actions at each step of a task and involves tracing out the possible solution path a student takes for a problem [10]. The model tracing method requires a large *bug library* (list of all possible paths students may take to end at a wrong solution) as well as a library storing the correct paths.

Constraint-based modelling however, represents problems to solve as a set of constraints [14, 15] a student must satisfy for the problem to be considered solved. This means that there is no need for a large bug library to represent incorrect knowledge, it only requires the correct knowledge to be satisfied by the students to solve a problem.

It has been shown that best student learning happens during one on one interaction with a tutor [5]. But this is not the case with today's education systems as it is too costly to carry out. ITS have been developed to provide the one on one interaction environment to students but can still be improved.

Disengagement in an ITS has been an area of interest for some time. Ideally, an ITS will adapt to when a user disengages from the system and re-engage the user back to the system. Many ITS today do not consider the process of

re-engaging a disengaged user.

Data mining provides a way in which we can find a suitable predictor from previous existing data to predict when future users will become disengaged. The predictor will then notify the ITS that the user will become disengaged and so the ITS can respond in a meaningful way to keep users engaged. ITS usually record information of users' actions without distracting the user from their task, which can be studied to help find a predictive model that can predict disengagement of users.

## 1.2 Motivation

The Intelligent Computer Tutoring Group (ICTG) is always looking for ways to improve students learning experiences. They have developed many ITS, one system being SQL-Tutor. SQL-Tutor teaches SQL, the most widely used database language. SQL-Tutor teaches the standard SQL and not a particular implementation of it and supports students' learning by asking students to solve problems that involve writing queries for various databases for a desired outcome.

The current version of SQL-Tutor does not provide any support for disengaged students. We are interested in whether a predictor and an intervention response can be implemented into SQL-Tutor in such a way that reduces disengagement. More specifically, we are interested in if a predictor that has been learned from existing data can accurately predict abandonment of problems in SQL-Tutor. Our focus is restricted to abandonment as there are many forms of disengagement, such as starting an unrelated conversation with a friend, abandoning tasks given by the tutor or closing the tutoring program, and trying to predict all forms of disengagement could be difficult.

### 1.3 Goals

Our goals were to find a predictor that can accurately predict disengagement of users using SQL-Tutor and also implement an intervention in SQL-Tutor to re-engage the disengaged users. The research questions are as follows:

- How can an accurate predictor be learned from existing data?
- What is the effectiveness of the developed intervention?

### 1.4 Report Structure

Section 2 summarises and discusses the background material relevant to this project and also mentions the related work done in this topic. It provides a brief overview of Intelligent Tutoring Systems and describes a short distinction between model tracing and constraint-based modelling and also outlines the particular ITS used in this project (SQL-Tutor) and gives information about the ITS used that is later mentioned or elaborated on. Section 3 reports the data involved in the project in depth, as well as what was derived from the data. Section 4 describes the experimental design and procedure. Section 5 reports the results from the experiment and explores the results to answer the research questions. Section 6 provides an overall discussion of this research project and states the limitations and future work. Finally, Section 7 summarises and concludes the report.

## 2 Background

Many intelligent tutoring systems use the ACT-R production system architecture [10] as a way to model students' correct and incorrect knowledge. According to ACT-R, there are two knowledge types, declarative and procedural knowledge. Declarative knowledge is knowledge acquired from in-

struction and/or reading with no commitment to how this knowledge can be utilized. Procedural knowledge is acquired through using declarative knowledge via problem-solving practice..

Model tracing is a technique to model students' knowledge by following the students step by step to figure out students' correct and incorrect knowledge [10]. An intelligent tutoring system that uses model tracing techniques will apply rules to every student action to guide the students toward the path to correct knowledge and away from the path to incorrect knowledge.

Constraint-based modelling [15] is a technique to model students' knowledge by checking that student answers satisfy defined constraints. Doing so removes the need to understand correct and incorrect knowledge of students but trades on this computational efficiency by ignoring the students problem-solving strategy.

## 2.1 SQL-Tutor

The project focuses on SQL-Tutor, a popular intelligent tutoring system developed by the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury, that supports students learning SQL (Structured Query Language). SQL-Tutor provides many problems for students to solve, a basic usage diagram of SQL-Tutor is shown in Figure 1.

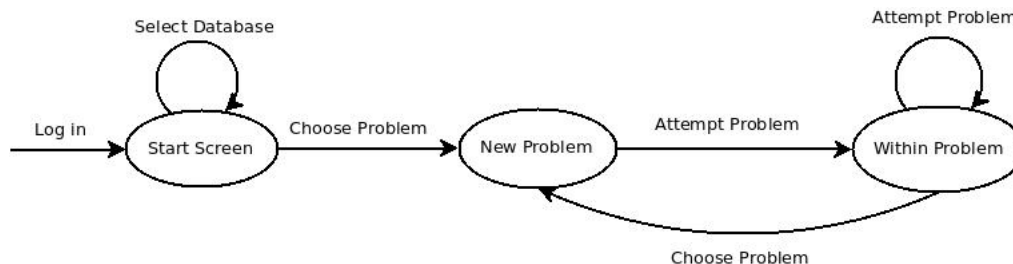


Figure 1: Basic usage of SQL-Tutor



Figure 1 shows how users typically use SQL-Tutor. Users can log out at any state shown in the diagram. SQL-Tutor provides many databases for users to choose from with each database containing many problems related to that database. The databases users can choose from are: books, cd-collection, company, computer-shop, cruises, library, movies, music, product, registration, shares, sponsors and woodwork. An example question from the books database is "List the titles of all paperbacks."

If the student's solution is wrong, SQL-Tutor provides feedback chosen to be suitable for the student based on the previous activity of the user. If the solution is correct, the student is notified that the attempt is correct and can progress to another problem.

The screenshot shows the SQL-Tutor interface in a web browser. The URL is `ictg.cosc.canterbury.ac.nz:8000/sql-tutor/respond-to-main`. The interface has a top navigation bar with buttons: **Change Database**, **New Problem**, **History**, **Student Model**, **Run Query**, **Help**, and **Log Out**. The main area is titled **Problem 28** and contains the question: "What is the name of customer number 8?". Below the question are input fields for the SQL query, with labels **SELECT**, **FROM**, **WHERE**, **GROUP BY**, **HAVING**, and **ORDER BY**. The current query is:
 

```
SELECT NAME
FROM CUSTOMER
WHERE CUSTOMER.ID = 8
```

 To the right of the query fields is a large text area for feedback, which currently contains the message: "All names in the SELECT clause must be attribute names from the tables listed in the FROM clause." Below the query fields are buttons for **Hint**, **Submit Answer**, and **Reset**. At the bottom of the interface is a section titled **Schema for the MOVIES Database** with a general description and a table listing attributes for **DIRECTOR**, **MOVIE**, **STAR**, and **CUSTOMER** tables.

Figure 2: The Problem interface of SQL-Tutor

Figure 2 shows the problem interface shown to users. The large box next to the input fields show the feedback from the tutor. The feedback box provides feedback based on the feedback level (located next to the submit button) the user has selected or if the user has not changed the feedback level themselves, then the tutor may give appropriate feedback based on the student model.

The actions made by the users are logged by SQL-Tutor, and include basic usage information such as the time a user logged in, the database that a user chose to work in, the problem received by a user, the attempt that the user made on a problem and so on. The logged information is kept in log files for each user. Here is an excerpt from a user's log file:

```
11:27:51 4/09/2013; Student's problem choice: 1
11:27:51 4/09/2013; set-new-problem sets help-level to 0
11:27:51 4/09/2013; drawing problem: 1, problem status: NEW
11:31:27 4/09/2013; responding: problem is 1 its status is NEW
11:31:27 4/09/2013; responding: also set help-level to 0, feed-
back=Simple Feedback
11:31:27 4/09/2013; Pre-process: Help Level 0; Feedback Option:
Simple Feedback; Database: cd-collection; Problem number: 1;
Their attempt: Select: *; From: artist; Where: ; Group by: ;
Having: ; Order by: ; Two-level-help?: NIL; Mode: 12
11:31:27 4/09/2013; Answer correct
11:31:27 4/09/2013; Now help-level is 0
11:31:27 4/09/2013; Post-process: Satisfied constraints: (146 141
380 674 93 471 350 48 131 94 104 126 106 103 105 101 100 108
107 78 98 91 90 87 86 801 800 81 80 71 70 155 673 95 55 10 351 75
74 6 701 528 700 525 365 364 668 667 3 2); Violated constraints:
NIL; Feedback level: 0
11:31:27 4/09/2013; Check point
```

Each line of the log file begins with the time and date of the action. This information is useful for finding time information such as how long a user spent on a question. There are also lines in the log files that may not be so useful for this project such as the "Check point" log.

## 2.2 Modelling Off-task Behaviour

Previous research of detecting student disengagement has been conducted for modelling and understanding students off-task behaviour in ITSs. Baker has

presented a model that automatically detects when students are off-task in Cognitive Tutor [3], a mathematics tutor developed by Carnegie Learning, and concluded that "off-task behaviour is associated with disliking computers, disliking mathematics, passive-aggressiveness, and lack of educational self-drive." The mentioned research was successful in finding an accurate predictor that can automatically detect when a student is off-task. However, this research is limited to model-tracing tutors as the research used data gathered from the Cognitive Tutor software, which is a model-tracing tutor, and does not generalise to constraint-based tutors.

The same research demonstrates that only log files containing basic student actions is enough to find an accurate student disengagement detectors.

### **2.3 Gaming in Constraint-Based Tutors**

Research into constraint-based tutors has focused on student gaming behaviours and Baker et al. have detected gaming the system of the SQL-Tutor [4]. Gaming students systematically take advantage of properties and regularities in SQL-Tutor, such as asking the tutor to provide the full answers repeatedly, to complete tasks. This research has shown that constraint-based tutors differ from other types of intelligent tutors for which gaming detectors have been developed in other fashions.

### **2.4 Disengagement Detection using Mouse Movement Information**

Research into detection of disengagement using time, performance and mouse movement features has also been investigated. This researched showed that the approach utilizing the mouse movement information outperforms a detection model that only utilizes time features and also outperforms a detection model that uses time and performance features together [6].

This research focused on utilizing the mouse movement features that were available and showed that a disengagement predictor learned using the mouse movement features outperformed the disengagement predictors that did not use the mouse movement features. This research does not investigate performance of disengagement predictors learned using constraint related features.

## **2.5 Disengagement Detection using Eye Tracking**

Research into detecting student disengagement using eye tracking methods has also been conducted in Gaze Tutor [7] which detected disengagement and boredom and developed interventions that used dialogs which responded to eye movement in an attempt to reorient students attention to the tutor. These prior research projects [3, 6, 7] have presented their own student disengagement detector models and the Gaze Tutor project has shown that gaze-reactive intervention did have the desired effect of directing students attention towards the tutor and produced a learning gain.

## **2.6 Detecting Quitting Behaviour in Reading**

Recent research in students' quitting behaviour [11] used sensor-free information from previous activities to predict quitting behaviours of students reading text. This research was able to generate accurate predictors of students' quitting behaviours of reading text of multiple scenarios (quitting on any page, quitting on page 2, quitting after page 1).

### 3 Predictor

One of the main goals of this project is to use previously collected data to produce a predictor that can accurately predict abandoning behaviour of users of SQL-Tutor. To produce such a predictor, there are many things to consider such as the data to produce a predictor from, the algorithms applied to the data and also aspects such as determining a good predictor.

The process of coming up with a predictor as defined by the Knowledge Discovery in Databases (KDD) [2] is as follows:

1. *Selection* - only relevant information is selected to be further processed.
2. *Pre-process* - important elements existing in the data is detected and processed into data that's more meaningful.
3. *Transformation* - transform data into data that can be used by a data mining tool.
4. *Data mining* - applies techniques to present patterns present in the data.
5. *Interpretation* - to look for knowledge in the patterns presented and to examine the cause of the patterns presented.

The *selection* stage for finding a predictor consisted of finding relevant datasets, i.e. looking at previous SQL-Tutor usage data.

The *pre-process* stage was performed by developing Python code to detect and gather the information required to find a predictor. This stage involved extracting the session elements from the log files and identifying the features. The log files may contain several *sessions* and each session may contain many *problems* that a user attempted or solved and each problem may contain several *submissions*. The relationship between log files, sessions, problems and submissions is illustrated in Figure 3.

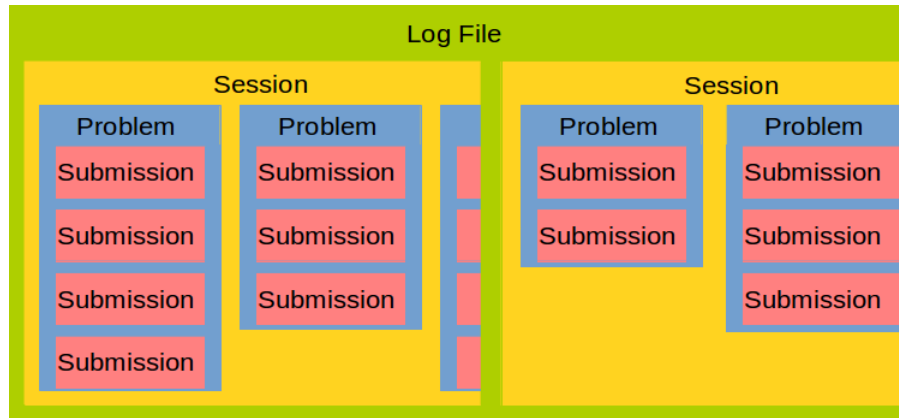


Figure 3: A simple illustration of a single log file and what it may contain

The extraction of session attributes was achieved by the extractor Python code. The code behaved as follows:

```

class Submission:
    attributes
    ...

class Problem:
    attributes
    ...
    Submissions=[]
    ...

def populate_submissions():
    ...

class Session:
    attributes
    ...
    Problems=[]
    ...

def populate_problems():
    ...
    populate_submissions()

```

```

def main():
    for log in logs:
        for session in log:
            session.populate_problems()

```

*Transformation* stage involved taking the pre-processed data and transforming it into the format that can be used by the data mining tool WEKA [9]. WEKA uses the Attribute Relation File Format (ARFF) extension for input files. An ARFF file contains a relation declaration, a features section and a data section. An ARFF file declares the relation and then the features at the beginning of the file, each feature is separated by a new line. After the features are declared, the data is declared in instances, each instance is separated by a new line and each instance contains the feature attribute observations horizontally separated by commas, the following is an example of the ARFF structure:

```

@relation weather

@attribute outlook sunny, overcast, rainy
@attribute temperature numeric
@attribute humidity numeric
@attribute windy TRUE, FALSE
@attribute play yes, no

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes

```

overcast,72,90,TRUE,yes  
overcast,81,75,FALSE,yes  
rainy,71,91,TRUE,no

The *data mining* stage required inputting the generated ARFF file as *training data* to WEKA and choosing appropriate algorithms to use to produce a predictive model. This could be done in many ways, i.e. through the provided GUI or java call in a terminal. WEKA provides many learning algorithms to train a predictive model and so careful considerations must take place to produce a predictor that performs the best. The outcome of running a learning algorithm in WEKA after supplying the training data is a predictive model as well as the summary of predictor accuracy and also a confusion matrix that show how well the predictive model performs.

Finally, *interpretation* looks at the detected patterns to see if they contain knowledge and so the results from data mining must be represented appropriately so that it can be examined thoroughly. This involved examining the models produced from data mining and interpreting the model's attributes (e.g. the tree size of a model produced from a decision tree algorithm) to see if there is any overfitting.

### 3.1 Datasets

A selection of different datasets were available to find a predictor. Overall, three different data sets were available (Table 1), each dataset consisted of log files of students using SQL-Tutor. Each dataset represents a different population.

The *2013* dataset consisted of log files that were recorded on fixed experiment where participants were required to solve a sequence of problems given by SQL-Tutor. The participants had no choice to abandon a problem as the experiment required the participants to solve the problems given to them sequentially. This dataset had no useful information about behaviours of users abandoning problems and was only used as sample log files to create



Table 1: Datasets used

Dataset	Description	Number of Logs
2013 Dataset	Data that was recorded of participants to a previous study using SQL-Tutor	48 (1376 instances)
2010 Dataset	Data that was recorded of COSC265 students using SQL-Tutor as part of their course-work	56 (1980 instances)
Addison-Wesley Dataset	Data recorded of users using SQL-Tutor through the Addison-Wesley’s web portal DatabasePlace <a href="http://www.aw-bc.com/databaseplace/">http://www.aw-bc.com/databaseplace/</a>	7519 (16541 instances)

the code to parse log files and was not further considered for this project.

The *2010* dataset was considered and used because the log files of this dataset contained data of COSC265 students using SQL-Tutor. This dataset was collected in a similar setting to our experiment. The 2010 dataset was collected from COSC265 students who could freely select problems in SQL-Tutor.

The *Addison-Wesley* dataset is a much larger dataset and was used because of the number of log files this dataset contains. The large number of log files means that this dataset contains information from a large number of users and so it contains many populations of users (students, beginners, experts...) which means that more cases of abandoning behaviour could be learned using this dataset.

## 3.2 Features

To learn the abandoning behaviours using previous data of SQL-Tutor usage, features must be extracted from the data. Features are attributes that can help learning algorithms learn the patterns inside large datasets and can be either numerical or categorical.

An iterative process of feature extraction can help determine what features

are useful. We decided that looking at the actions leading up to an abandon action is important, so we looked at three different categories of features: *current problem features*, *previous problem features* and *cumulative problems features*.

The *current problem features* provide information regarding the current problem that a user is attempting and could potentially abandon. The current problem features could provide information about what users did during solving a problem that lead to abandoning the problem and consisted of 10 features listed in Table 2.

There were features that could be directly extracted from the log files such as problem complexity or the help level requested by the user for a problem but there were also many features that required calculations using the low level data from the log files. The calculated features for *current problem features* were:

time from start = latest submit time - start time for the current problem

time from previous submission = latest submit time - previous submit time

time since session start = latest submit time - session start time

decreased violated constraints = true if previous submission violated constraints > current submission violated constraints, else false

submission time difference = ((previous submit time - previous previous submit time) - (current submit time - previous submit time))<sup>2</sup>

The *previous problem features* looked at information regarding the previous problem that the user attempted or solved. These features could provide important pattern information of what users did in the previous problem they attempted before they abandon the next problem. 16 features for this categories were derived and is listed in Table 3.

As with the *current problem features*, the *previous problem features* contained many calculated features for the previous problem, such as:

Table 2: Current Problem Features

Feature	Description
violated constraints	number of violated constraints
satisfied constraints	number of satisfied constraints
complexity	the complexity of the current problem
help level	the help level the user asked for for that submission
time from start	time since when the user first received the problem
time since previous submission	time since the previous submission of the problem
submission number	number of attempts made at the current problem
time since session start	time since when the user started the session/logged in
decreased violated constraints	a boolean value, true if the latest submission has decreased violated constraints than the previous submission, else false
submission time difference	the squared difference value in time between the previous two submissions

first submit time = submit time of first submission - problem start time

time taken = last submit time - problem start time

average submission time =  $\text{sum}(\text{difference in submission times}) / \text{number of submissions}$

time for the latest submission = latest submit time - second latest submit time

The *cumulative features* looked at the information of the session that the users are currently working in. These features are described in Table 4. These features looked mainly at the extremes and the averages of what was achieved by the user in the session.

Having the previous problem features to emphasize learning the actions leading up to abandonment meant that these features can not be found for when a user first enters SQL-Tutor to begin their lesson. This also meant that

Table 3: Previous Problem Features

Feature	Description
first submit time	the time it took the user to submit their first attempt from when given the problem
time taken	time taken to complete the problem
completed	true for if the previous problem was completed
submission number	the number of submissions made for the problem
maximum violated constraints	the maximum number of violated constraints
number of wrong submissions	the number of submissions with one or more violated constraints
average submission time	the average time between submissions
latest submit time	the latest time it took to submit from the previous submission
stdv of submit time	the standard deviation of the submit times
maximum submit time	the maximum time it took to make a submission
minimum submit time	the minimum time it took to make a submission
same database	true if using the same database as the current problem
complexity	the complexity level of the problem
feedback option	the number of different feedback options the user received
same submissions	the number of same submissions the user made
time since session start	the time since the session began until the beginning

making a prediction to see if a user will abandon their first problem or not will have missing values for the previous problem features. This may not be ideal because predicting while missing the previous problem features is making a prediction based on just the current and first problem that the user is attempting. To ensure that the previous problem features are utilised, the first two problems attempted or solved in each session was omitted from feature extraction except for the cumulative features extraction.

Table 4: Cumulative Features

Feature	Description
problems attempted	the total number of problems attempted during the session so far
problems completed	the total number of problems completed during the session so far
database changes	the number of times a user changes databases during the session
time for completion <sup>†</sup>	time it took to complete problems
time between submissions <sup>†</sup>	time it took between consecutive submissions
number of submissions <sup>†</sup>	the number of submissions for each problem
violated constraints <sup>*</sup>	number of violated constraints
problem complexity <sup>**</sup>	the problem complexities encountered

<sup>†</sup> calculated average, standard deviation, maximum, minimum

<sup>\*</sup> calculated average and maximum

<sup>\*\*</sup> calculated average, standard deviation, maximum

### 3.3 Choosing the Best Predictive Model

Multiple predictors were studied before deciding on a predictor to use for the experiment. Choosing the best predictive model meant that "best" had to be defined for a predictive model. A good predictor to be used in the study will be able to predict both the abandonment and non-abandonment cases of users with high accuracy in all situations. However, this is not easily realised because of many trade-offs the process of producing a predictor presents.

## Cross Validation

Cross validation is a model validation technique to show the results from a model will generalise to an independent dataset [1] and is commonly used where the goal is prediction and an estimate of how accurate a predictive model will perform in practice is required. In a prediction problem, a model is usually given a training set and a test set. The training set is used by the model to learn what is required to make predictions and the test set is unknown data that the model tests its predictions on to give an accuracy measure.

The goals of cross validation is to provide a validation dataset during the model training phase in order to limit problems such as overfitting and also to give insight to how the model will generalize in practice.

One round of cross validation involves splitting up the given dataset into two complementing subsets, training the model with a subset and testing the trained model with the other subset. A k-fold cross validation will involve k rounds of cross validation, each round producing different subsets for training and testing.

Some common types of cross validation are 10-fold cross validation and leave-one-out cross validation. 10-fold cross validation has 10 rounds of cross validation and leave-one-out cross validation has  $N - 1$  rounds of cross validation where  $N$  is the number of instances provided in the input dataset.

## Predictive Models from Datasets

The 2010 dataset contained 1980 instances in total, of which 227 were abandonment cases leaving 1753 non-abandonment cases. Applying the 10-fold cross validation technique on the 2010 dataset using the J48 tree algorithm [9] produced a predictor ( $P_{2010,J48}$ ) with accuracy measures shown in Table 5.

Table 5: Accuracy measures for  $P_{2010,J48}$  predictor

		<b>Predicted</b>	
		Abandoned	Not abandoned
<b>Real</b>	Abandoned	194	33
	Not abandoned	24	1729
Correctly Classified Instances		1923	97.12%
Incorrectly Classified Instances		57	2.88%

Table 5 shows that  $P_{2010,J48}$  will be very accurate when predicting the abandonment and non-abandonment cases of similar future data.

The Addison-Wesley dataset contained 16541 instances in total, of which 7854 were abandonment cases leaving 8687 non-abandonment cases. Applying the 10-fold cross validation technique on this dataset using the J48 tree algorithm produced a predictor ( $P_{AW,J48}$ ) with accuracy measures shown in Table 6.

Table 6: Accuracy measures for  $P_{AW,J48}$  predictor

		<b>Predicted</b>	
		Abandoned	Not abandoned
<b>Real</b>	Abandoned	6992	862
	Not abandoned	868	7819
Correctly Classified Instances		14811	89.54%
Incorrectly Classified Instances		1730	10.46%

Similar to  $P_{2010,J48}$ , it is suggested that  $P_{AW,J48}$  will be very accurate when predicting the abandonment and non-abandonment cases of similar future data. One interesting observation is the ratio of abandonment and non-abandonment cases in both the 2010 and Addison-Wesley datasets. The

proportion of abandons to not-abandons for the 2010 dataset is much lower than the abandons to not-abandons for the Addison-Wesley dataset, 11% and 47% respectively.

I then wanted to see how well  $P_{2010,J48}$  could predict the Addison-Wesley dataset and also how well  $P_{AW,J48}$  could predict the 2010 dataset. I wanted to see this result because I wanted to know if either predictors were learned using a dataset that can represent another.

Table 7:  $P_{2010,J48}$  with Addison-Wesley dataset as test set

		<b>Predicted</b>	
		Abandoned	Not abandoned
<b>Real</b>	Abandoned	2236	5618
	Not abandoned	609	8078
Correctly Classified Instances		10314	62.35%
Incorrectly Classified Instances		6227	37.65%
Correctly Classified Abandons		2236 (7854)	28.47%

Table 8:  $P_{AW,J48}$  with 2010 dataset as test set

		<b>Predicted</b>	
		Abandoned	Not abandoned
<b>Real</b>	Abandoned	143	84
	Not abandoned	807	946
Correctly Classified Instances		1089	55%
Incorrectly Classified Instances		891	45%
Correctly Classified Abandons		143 (227)	63%

Tables 7 and 8 show the results of training the model using the J48 tree algorithm using either the 2010 dataset or the Addison-Wesley dataset and



testing the model on the other dataset. These results were expected to have poor accuracy because the training and testing data are two different datasets. What was interesting was that  $P_{AW,J48}$  was able to correctly classify 63% of the abandons in the 2010 data, whereas  $P_{2010,J48}$  was only able to correctly classify 28.47% of abandons in the Addison-Wesley data. This suggested that having a large number of abandons in the training data may produce a predictor that can better classify abandons.

I then looked at how combining the datasets would affect the accuracy measures of a model trained using the combined datasets. The predictor learned using the J48 tree algorithm with training set being the combined datasets of the 2010 dataset and Addison-Wesley dataset ( $P_{combined,J48}$ ) showed accuracy measures shown in Table 9.

Table 9:  $P_{combined,J48}$ , predictor learned using combined dataset

		Predicted	
		Abandoned	Not abandoned
<b>Real</b>	Abandoned	7124	957
	Not abandoned	945	9495
Correctly Classified Instances		16619	89.73%
Incorrectly Classified Instances		1902	10.27%

The combined dataset had in total, 18521 instances, of which 8081 were abandonment cases leaving 10440 non-abandonment cases. The results from the 10-fold cross validation technique shows that  $P_{combined,J48}$  is only slightly worse performing than  $P_{2010,J48}$  and  $P_{AW,J48}$ .

## Exploring Algorithms

Five algorithms were explored including the J48 decision tree algorithm, they were:

Table 10: Comparing Various Algorithms using the combined Dataset

Dataset	Results	DT	BN	L	AB	CR
Combined	%correct	<b>89.73</b>	77.44	79.67	75.56	87.80
	Kappa	<b>0.79</b>	0.52	0.58	0.49	0.75
	TPR	<b>0.88</b>	0.58	0.71	0.57	0.85
	FPR	0.09	<b>0.08</b>	0.13	0.10	0.10

- DT - J48 decision tree algorithm
- BN - Bayesian net K2 search algorithm
- L - Logistic classification algorithm
- AB - AdaBoostM1 DecisionStump classifier
- CR - Classification via Regression M5P tree classifier

The results of each algorithm for the combined dataset running 10-fold cross validation is shown in Table 10.

The predictor learned using the J48 decision tree algorithm on the combined datasets is shown to be a more accurate predictor on future similar data than the other four algorithms and so this predictor will be used for the experiment.

To summarise, the predictor chosen to be used in our experiment ( $P_{final}$ ) was learned using the J48 decision tree algorithm on the combined dataset. The tree size of  $P_{final}$  was 1565 nodes with 783 nodes being leaf nodes.

## 4 Experiment

### 4.1 Goals and Hypotheses

The main objective of this research project is to deploy the predictor found in Section 4 and to apply an intervention when the predictor predicts if a user will abandon a problem in SQL-Tutor. This investigation is carried out by modifying SQL-Tutor. The modification will include designing and implementing an experimental interface that predicts problem abandonment and produces an intervention when the user is predicted to abandon a problem.

To summarise, the goals of this research project are:

1. Evaluate the predictor’s accuracy through experimentation
2. Evaluate the effectiveness of the interface that produces an intervention

Correspondingly, my hypotheses are:

1. The predictor discussed in Section 4 will be accurate for the experiment.
2. The experimental interface that produces an intervention will be effective at keeping users engaged in the system.

### 4.2 Intervention Design

There are many possibilities for intervention design to keep users engaged in SQL-Tutor. As mentioned before, one possibility is to show motivational messages to increase the motivation of users using the system to keep them engaged.

A motivational intervention is desirable to apply when a user is predicted to abandon a problem in SQL-Tutor, but there are many ways to motivate a

user, for example, showing the progress that the user has achieved or showing a message to assure the user is learning. We decided to use motivational messages that encouraged users to stay engaged within the system by challenging users to complete the problem, giving a motivational message to the users or by giving encouraging details of SQL as outlined in Table 11.

In total, 11 motivational messages were implemented to be shown, these messages are outlined in Table 11. The messages were given at random without repeat when a user was predicted to abandon a problem. If a user would receive all 11 messages, the messages would cycle and be given at random again.

Table 11: Motivational messages shown to users who were predicted to abandon a problem

Message
You can complete this problem!
Keep going!
Completing this problem means you're becoming an expert.
Learning SQL means learning the most widely used database language.
SQL is a powerful tool for a programmer.
SQL is a great language to learn.
SQL is a powerful fourth generation language for accessing data.
With effort, you can master SQL.
Complete the problem to confirm your knowledge.
Complete this problem by thinking of each clause carefully.
Use the feedback to help you complete this problem.

The aspect of when to present the intervention was thought about. We decided that a delayed intervention upon a users' submit action would be less irritating. A delayed intervention would be less irritating because an

intervention is not what users expect upon submitting an answer to the system and so a delayed intervention would give users the opportunity to reorient themselves before receiving an intervention. The delay was set to 3 seconds.

Another important aspect considered when presenting an intervention to a user was how to present the intervention to the user. Using the feedback box of SQL-Tutor may not be the best place to display the intervention messages as this text box is mostly used for feedback of the users submissions and so having messages unrelated to their submissions may confuse or annoy the users.

As much as the general public dislikes pop-up boxes, this is a great means to display the motivational messages to the users in an eye-catching way as well as receive users' feedback about the intervention by asking them if they agree with the prediction that they were going to abandon the current problem or not (Figure 4).

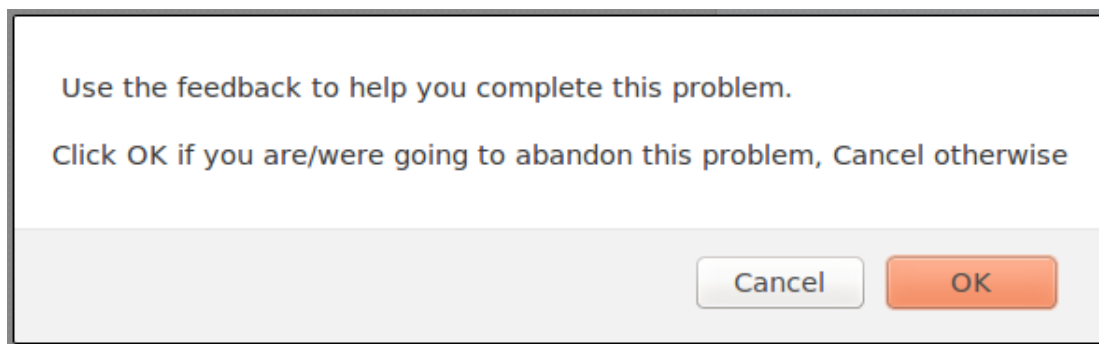


Figure 4: Intervention message shown in Firefox

### 4.3 Procedure

To test the predictor's accuracy and to test the effectiveness of the intervention, the experiment was divided into two phases. In Phase 1, there is no

noticeable change in the interface. In Phase 2, the student receives an intervention message through a pop up dialog if they were predicted to abandon the latest problem.

SQL-Tutor is available to the participants through the web browser. The SQL-Tutor web client is written in common-lisp. Implementations of the experimental interface was accomplished by modifying the SQL-Tutor source code.

## **Phase 1**

In Phase 1 consisted of participants used a version of SQL-Tutor that made predictions on abandoning and not abandoning problems in the background without any visible changes shown to the participants. This was accomplished by calling the extractor Python code in the common-lisp code to calculate the features of the latest submission and supplying the new feature attributes to the predictor. Once the predictor is supplied with the feature attributes, an abandon or not-abandon prediction is made inside the extractor code. The extractor code returns the output back to the SQL-Tutor code. The output from the extractor code is logged into the participant's log file to be later analysed. Figure 5 shows the process more clearly.

We decided that 40 minutes is an acceptable amount of time for participants to spend in phase 1 and to collect enough data to evaluate the predictor accuracy. This was based on a 100 minute scheduled lab session and so if a participant spends the full duration of the scheduled lab session using SQL-Tutor, they would record 40 minutes of data in the control phase and 60 minutes of data in the experimental phase.

## **Phase 2**

Phase 2 consisted of participants using a version of SQL-Tutor that presented an intervention with one of the messages from Table 11. The procedure is

similar to that of Phase 1 but an intervention is applied if the prediction is that the participant is abandoning the current problem (Figure 5).

Although the scheduled lab session for COSC265 is for 100 minutes, we anticipated that most participants won't stay for the whole duration of the lab session and so the length of Phase 1 is 40 minutes and the rest of the session will be Phase 2.

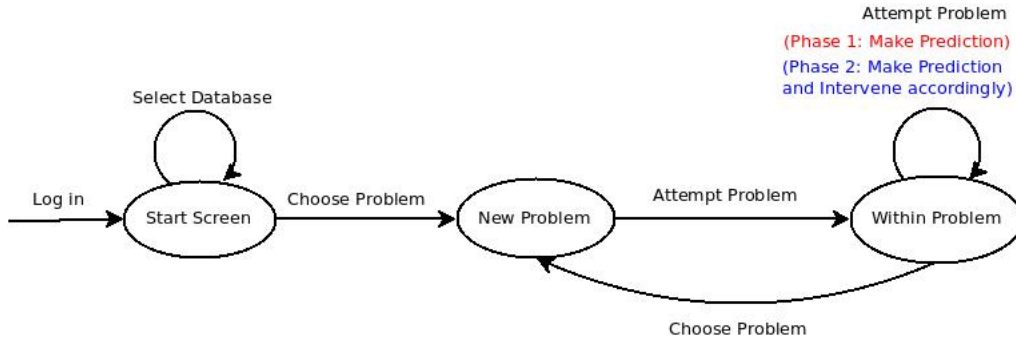


Figure 5: SQL-Tutor usage of the Experiment

#### 4.4 Participants and Apparatus

The participants are COSC265 students using the University of Canterbury's lab computers to participate in the experiment during one of their scheduled lab sessions. The lab computers provide firefox and chromium for the web browsers and all students are asked to use the lab machine for the experiment instead of their own personal computers. The participants were asked to sign a consent form for their consent to partake in this experiment. Any participant who did not wish to participate in this experiment were removed from any analysis.

The experiment was approved by the Human Ethics Committee. The application and the approval are given in Appendix A.

## 5 Results

The experiment showed interesting results. 49 students participated in the experiment. The data collected about three participants were removed from further analysis because they have interacted with SQL-Tutor for a short time. Two participants used the system for less than 10 minutes. Another participant behaved in an abnormal manner by spending less than 10 minutes during Phase 1 but spending over 40 minutes in Phase 2 and thus, was removed from further analysis.

There was also participant s7, who spent a short time in phase 2 (0.4 minutes). We have kept data collected from this participant for Phase 1, but have not used the data for Phase 2.

The participants' demographic information from questionnaires is shown in Table 12.

Table 12: Demographic Information

	Gender	
	Male	Female
Age Group	18-23	33      7
	24-29	2      0
	30-35	2      0

44 participants completed the questionnaires. It also shows that the majority of the participants were male of ages between 18 and 23.

Participants spent an average of 35.61 minutes in Phase 1. A few students spent more than the predetermined time in Phase 1 of 40 minutes due to logging out of the system during the experiment and recorded a maximum time spent in Phase 1 at 81.82 minutes. The overall results of the experiment are shown in Tables 13 and 14.



Table 13: Overall results for Phase 1 of the Experiment of 46 participants

	Maximum	Minimum	Average	St. Dev.
Duration (minutes)	81.82	17.9	35.61	10.92
# Attempted Problems	19	4	9.94	4.16
# Completed Problems	18	4	9.09	4.16
# Submissions	128	7	40.63	21.74
Not-abandon Predictions <sup>1</sup>	106	7	36.59	15.52
Abandon Predictions <sup>2</sup>	26	0	4.07	5.99

<sup>1</sup> The number of times a participant was predicted to not abandon by the predictor

<sup>2</sup> The number of times a participant was predicted to abandon by the predictor

To further explain the results shown on Table 13, the maxima for the *Attempted* and *Completed* problems were by one participant but this participant did not have the highest number of *Submissions* meaning that another participant submit more attempts but completed less problems. Also, the participant who has the highest number of *Submissions* also has the highest number of *Not-abandon Predictions* but another participant has the maximum value for *Abandon Predictions* meaning that the latter participant was predicted to abandon more than the former participant with less submissions.

38 out of 46 participants stayed in the system for over 40 minutes to enter Phase 2 of the experiment where they would receive an intervention if they were predicted to abandon a problem. The overall results of Phase 2 show that on average, participants spent 26.14 minutes in Phase 2 (Table 14).

Table 14: Overall results for Phase 2 of the Experiment

	Maximum	Minimum	Average	St. Dev.
Duration (minutes)	61.9	3.17	26.14	14.57
# Attempted Problems	17	1	5.29	3.56
# Completed Problems	17	0	4.63	3.66
# Submissions	62	3	22.76	13.42
Not-abandon Predictions <sup>1</sup>	50	3	18.68	11.21
Abandon Predictions <sup>2</sup>	15	0	4.08	4.65

<sup>1</sup> The number of times a participant was predicted to not abandon by the predictor

<sup>2</sup> The number of times a participant was predicted to abandon by the predictor

## 5.1 Predictor Accuracy

One goal of this project was to produce an accurate predictor to predict if users using SQL-Tutor would abandon a problem. To look at the predictor accuracy in depth, we separate the data of Phase 1 from the logs and generate a test data set that can be analysed using WEKA to evaluate the predictor used for the experiment. Separating the Phase 1 information from the log files was performed manually for the 46 log files by removing the lines that corresponded to Phase 2 information. The edited "Phase 1 only" log files can be transformed to an ARFF file using the extractor code discussed previously.

Only Phase 1 data can be used to evaluate the predictor accuracy using evaluation through WEKA because Phase 2 invokes a change in behaviour by applying an intervention.

The results from providing the new Phase 1 data as the test set to the predictor used for the experiment are shown in Tables 15 and 16.

Table 15: Confusion matrix on test data

	Predicted		
	Abandoned	Not abandoned	
Real	Abandoned	24	62
	Not abandoned	151	828

Table 15 shows what outcome each instance is as well as what outcome each instance was predicted to be. It shows that only 24 from 86 abandoned instances were correctly predicted as abandon cases. This is a True Positive Rate of 0.28. The not abandoned cases were predicted with a much higher accuracy with 828 out of 979 instances being correctly predicted as not abandoned cases so a True Negative Rate of 0.85.

Table 16: Summary of predictor accuracy

Correctly Classified Instances	852	80%
Incorrectly Classified Instances	213	20%
Kappa statistic	0.08	
Total Number of Instances	1065	

Furthering on the results shown in Table 15, Table 16 shows that the percentage of correctly classified instances is 80%. This value is high because the low True Positive Rate is diluted by the high True Negative Rate and also the fact that there are 979 not abandoned cases and only 86 abandoned cases in the test data.

Table 16 also shows the Kappa statistic value to be 0.0848 and is close to 0, meaning that the predictor only performs marginally better than guessing knowing the abandoned/not abandoned ratio.

During Phase 2, participants had the opportunity to give subjective feedback when they received an intervention. When participants were shown an intervention they were asked to provide feedback about whether they felt that they were going to abandon or not as described in Section 4.3. The results

from the subjective intervention feedback show that from the 38 participants that entered Phase 2, 24 participants received at least one intervention and gave feedback on the intervention pop up. The ratio between agreeing with the intervention and disagreeing with the intervention is close to 1:4 with the rate at which participants agreed with receiving an intervention was 0.202 meaning that participants felt that they were receiving the interventions wrongfully.

The information from questionnaires can provide more insight into the accuracy of the predictor from the participants' subjective feedback. The participants were asked to rate the accuracy of the predictor in the questionnaire. When asked if they felt that they were going to abandon a problem when an intervention message was shown, 5 participants answered yes while 30 participants answered no.

## 5.2 Effectiveness of Intervention

An independent-samples t-test was conducted to compare the problem completion rates in Phase 1 and Phase 2. There was no significant difference of problem completion rates in Phase 1 ( $Mean = 0.90$ ,  $SD = 0.11$ ) and Phase 2 ( $Mean = 0.81$ ,  $SD = 0.27$ );  $t(37) = 0.98$ ,  $p = 0.33$ .

These results suggest that there was no difference in the completion rates of problems in Phase 1 and 2. The mean of the problem completion rate is lower in Phase 2 than 1. This could be because the problems attempted by participants in Phase 2 are more difficult since the participants have been using the system for longer than Phase 1.

To see if participants faced more difficult problems in Phase 2, we looked at the problem complexities of the problems attempted by the participants in each phase. A total of 419 problems were attempted in Phase 1, and 238 in Phase 2. The problem complexity distribution of each phase is shown in Figure 6.

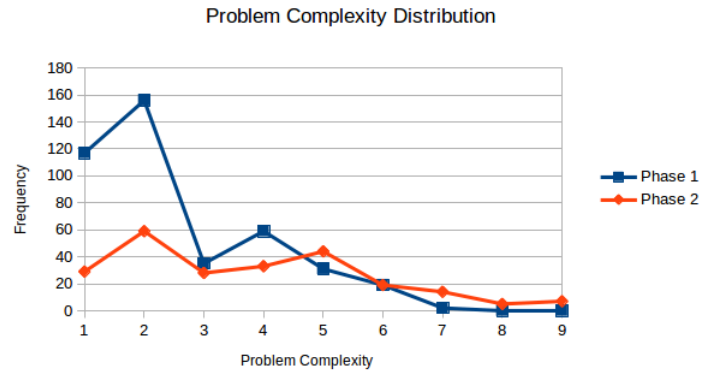


Figure 6: Problem Complexity Distribution Chart

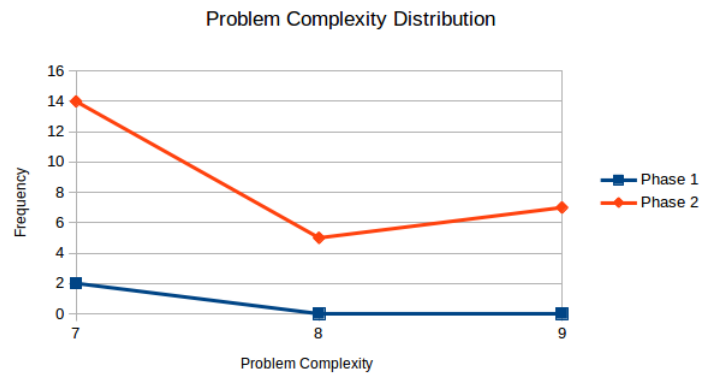


Figure 7: Difficult Problem Complexity Distribution Chart

Looking at only the difficult problems with complexities of 7 or over, it is shown in Figure 7 that the more difficult problems with complexities of 8 or over were only attempted in Phase 2.

Participants were also asked if the interventions were effective at keeping them engaged in SQL-Tutor in the questionnaires, to which 18 participants answered yes and 19 participants answered no. When asked if they felt that they were wrongfully shown an intervention, 9 participants answered yes and 18 participants answered no.

The results from the questionnaires suggest that the interventions were not accurately targeted and were shown to participants who were not going to abandon the problem when they received an intervention as 30 from 35 participants felt that they disagreed when shown an intervention but nonetheless, the receiving an intervention kept participants engaged as 18 from 37 participants answered that receiving the intervention was effective at keeping them engaged in SQL-Tutor.

When the participants were asked if a tutor that predicts abandonment and intervenes to keep students engaged in SQL-Tutor, 28 participants answered yes, 5 answered no and 5 answered maybe. Some comments were:

"Yes, in moderation. If used too much it will have no effect."

"Yes, though it depends on the method of intervention. A blatant intervention could potentially frustrate a student further."

"How it intervenes is more important to consider such as suggesting alternate questions more refined by the errors"

"No otherwise the student will waste time not making ground"

"Probably but it can get very annoying after a few"

Participants also commented about the intervention method. In general, participants mentioned that the interventions get annoying after a while and disliked the pop up dialogs. Some comments regarding the intervention methods were:

"The actual tutor is nice, but the messages are just annoying. When is the last time you thought Oh goodie! A pop-up! Never."

"After I'd received 3-4 intervention messages I found myself not reading them and clicking cancel automatically."

"Message gets annoying after a few times."

The results from the questionnaire indicate that most participants agreed that an ITS that predicts when students abandon problems and intervenes to

keep students engaged in the system is a good idea but careful considerations must be made to intervene at the right time and be presented in a way such that it does not distract students.

## 6 Discussion

Looking at Table 15 and 16 the predictor that was used for the experiment showed an accuracy rate of 80% but showed a low true positive rate at 0.28 and a high true negative rate at 0.84. Also the predictor showed a low Kappa statistic value at 0.08 indicating that the predictor was only slightly better than guessing the predictions. These results indicate that the predictor was weak at predicting abandonment cases but was accurate at predicting the non-abandonment cases.

A possible reason for the weak predictive power of the abandonment cases could be because of the low number of abandoned problems during the experiment where 86 from 1065 cases were abandons. The low number of abandons in the experiment differs with the high number of abandons in the dataset that the predictor was learned from where 8081 from 18521 cases were abandons and the low true positive rate suggests that the abandonment behaviours of the experiment was not accurately learned from the combined dataset used to produce the predictor.

The subjective feedback given by the participants when an intervention message was shown during Phase 2 also indicate that the predictor was not accurate with the abandonment predictions because the rate at which participants agreed with receiving an intervention message was 0.202.

One main goal of this research was to use previously collected data to produce a predictor that can accurately predict future abandonment behaviour in SQL-Tutor. The results from the experiment show that the predictor used for the experiment, which was learned using the J48 decision tree algorithm on a combined dataset consisting of the 2010 dataset and the Addison-Wesley dataset, did not accurately predict the abandonment of the experiment par-

ticipants.

The results of the effectiveness of the intervention used in the experiment suggest that the intervention messages were effective at engaging for some participants and annoying for other participants. The t-test conducted to compare the problem completion rates in Phase 1 and 2 indicate that there were no significant difference in participants completing problems in each phase. The problem complexity in each phase was investigated to explain the lower mean rate of problem completion in Phase 2 and showed that the participants

## 7 Conclusion

It was found that a predictive model learned using the J48 decision tree algorithm on a dataset that was a combination of SQL-Tutor usage in a university lab setting and SQL-Tutor usage by online users could not accurately predict the abandonment of problems in SQL-Tutor.

Investigations took place to find a predictive model using previously recorded data of SQL-Tutor usage, that could accurately predict the abandoning behaviours of new students using SQL-Tutor. The predictor selected to be used for the experiment showed superior performance of accuracy over other predictors and was learned using the J48 decision tree algorithm on a dataset that was a combination of SQL-Tutor usage in a university lab setting and SQL-Tutor usage by online users.

An experiment was designed to test the predictor and an intervention method which involved displaying a motivational message to the participants via a pop up dialog. The intervention messages were shown to participants who were predicted to abandon a problem by the predictor and their response was recorded. The experiment was separated into two phases, the first phase would make predictions in the background with no visible changes to the interface that the participants sees and the second phase would display an intervention message if a participant was predicted to abandon.



The data that was retrieved from Phase 1 of the experiment as well as participants' subjective feedback from questionnaires showed that the predictor used for the experiment could not predict the abandonment cases with high accuracy but could predict the non-abandonment cases with high accuracy. And the data retrieved from Phase 2 of the experiment showed that the pop up dialog intervention method showing a motivational message was an effective engaging intervention for some participants, while other participants found it annoying.

## **7.1 Limitations**

The dataset used to learn the predictor used for the experiment was shown to be not representative of the data recorded of the experiment. If a predictor was learned using a dataset that represented the experimental data, the effects of the intervention method could be better researched.

The intervention method was set to be motivational messages using a pop up dialog to display these messages. Other ways to display these messages could have had an effect on the results.

## **7.2 Future Work**

Much of the recorded data from the experiment remains un-investigated. The data could be further analysed to produce more results about the accuracy of the predictor as well as the effectiveness of the intervention method.

This project researched a static predictor, but a dynamic predictor could be interesting to research where a predictor is produced for each student that dynamically changes with the student's actions.

# Bibliography

- [1] Cross validation. In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, pages 441–441. Springer Netherlands, 2008.
- [2] Ana Azevedo and Manuel Filipe Santos. Kdd, semma and crisp-dm: a parallel overview. In *IADIS European Conf. Data Mining'08*, pages 182–185, 2008.
- [3] Ryan S.J.d. Baker. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1059–1068, New York, NY, USA, 2007. ACM.
- [4] RyanS.J.d. Baker, Antonija Mitrovi, and Moffat Mathews. Detecting gaming the system in constraint-based tutors. In Paul De Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 267–278. Springer Berlin Heidelberg, 2010.
- [5] Benjamin S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, 1984.
- [6] S. Cetintas, Luo Si, Yan Ping Xin, and C. Hord. Automatic detection of off-task behaviors in intelligent tutoring systems with machine learning techniques. *Learning Technologies, IEEE Transactions on*, 3(3):228–236,

July 2010.

- [7] Sidney D’Mello, Andrew Olney, Claire Williams, and Patrick Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies*, 70(5):377 – 398, 2012.
- [8] Davide Fossati, Barbara Di Eugenio, Christopher W Brown, Stellan Ohlsson, David G Cosejo, and Lin Chen. Supporting computer science curriculum: Exploring and learning linked lists with ilist. *Learning Technologies, IEEE Transactions on*, 2(2):107–120, 2009.
- [9] Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, IanH. Witten, and Len Trigg. Weka. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. Springer US, 2005.
- [10] Kenneth R. Koedinger, John R. Anderson, William H. Hadley, and Mary A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, pages 30–43, 1997.
- [11] Caitlin Mills, Nigel Bosch, Art Graesser, and Sidney DMello. To quit or not to quit: Predicting future behavioral disengagement from reading patterns. In Stefan Trausan-Matu, KristyElizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, volume 8474 of *Lecture Notes in Computer Science*, pages 19–28. Springer International Publishing, 2014.
- [12] Antonija Mitrovic. Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2):39–72, 2012.
- [13] Antonija Mitrovic and Brent Martin. Evaluating the effects of open student models on learning. In Paul De Bra, Peter Brusilovsky, and Ricardo Conejo, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 2347 of *Lecture Notes in Computer Science*, pages 296–

305. Springer Berlin Heidelberg, 2002.
- [14] Antonija Mitrovic and Stellan Ohlsson. Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems (COMSIS)*, 3(1):1–22, 2006.
- [15] Stellan Ohlsson. Constraint-based student modeling. In *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, volume 125 of *NATO ASI Series*, pages 167–189. Springer Berlin Heidelberg, 1994.
- [16] Kurt Vanlehn. The behavior of tutoring systems. *Int. J. Artif. Intell. Ed.*, 16(3):227–265, 2006.

## 8 Appendix

## Questionnaire

What is your age? (*Please circle*)

18-23      24-29      30-35      36-41      42-47      48+

What is your gender? (*Please circle*)

Female      Male

How much SQL knowledge did you have before learning about it in COSC265?  
(*Please circle*)

None      very little      little      moderate      lots      expert

How many intervention messages did you receive from the tutor? If you can't remember the exact number, approximately how many?

---

If you've received any intervention messages from the tutor, did you feel as if you were going to abandon the given problem when the message was given?

Yes    No

If yes, please describe your feelings before receiving a message

---

Do you think that receiving a message was effective at keeping you engaged in SQL-Tutor?

Yes    No

If yes, how so?

---

Did you at any time feel as if the tutor wrongfully gave you a message?

*Please describe*

---

Do you think that a tutor that predicts when a student will abandon a problem and intervene to keep the student engaged in the system is a good idea?

*Please describe*

---

Any other comments you would like to make?

*(Please write in the space below)*

Department of Computer Science and Software  
Engineering  
Telephone: +64 322 305 7298  
Email: [jkh601@uclive.ac.nz](mailto:jkh601@uclive.ac.nz)  
Date: 03-09-2014

## **Identifying and responding to disengagement in a constraint-based intelligent tutoring system**

### **Information Sheet for Study Participants**

This study is conducted by Jin Kwang Hong, a student at the University of Canterbury undertaking Honours in Computer Science. The purpose of the research is to investigate whether a predictor learned from previous data of students using SQL-Tutor can accurately predict new students abandoning a problem and also see if an intervention applied keeps students engaged.

Your involvement in this project will be to use a version of SQL-Tutor during a lab session which has a predictor and intervention integrated and will display a message when the predictor predicts whether you will abandon the given problem. You will also be asked to fill in a questionnaire at the end of the lab session.

There is no subsequent action you need to take on completing this study.

You may receive a copy of the project results by contacting the researcher at the conclusion of the project.

Participation is voluntary and you have the right to withdraw at any stage without penalty. If you withdraw, I will remove information relating to you. This will become impossible once the data has been analysed and results collated.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in the investigation: your identity will not be made public without your prior consent. To ensure anonymity and confidentiality, your name and details will not be stored with the data gathered from the study. The only people who will have access to the data will be members of the Intelligent Computer Tutoring Group (ICTG), who are a small group of staff from the Department of Computer Science and Software Engineering.

The project is being carried out as part of COSC460 Research Project course at the University of Canterbury by Jin Kwang Hong under the supervision of Prof Tanja Mitrovic and Dr. Kourosh Neshatian, who can be contacted at [tanja.mitrovic@canterbury.ac.nz](mailto:tanja.mitrovic@canterbury.ac.nz) and

kourosh.neshatian@canterbury.ac.nz respectively. They will be pleased to discuss any concerns you may have about participation in the project.

This project has been reviewed and approved by the University of Canterbury Human Ethics Committee, and participants should address any complaints to The Chair, Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (Human-ethics@canterbury.ac.nz).

If you agree to participate in the study, you are asked to complete the consent form and return it to Jin Kwang Hong upon undertaking the study.

-Jin



**UNIVERSITY OF CANTERBURY  
HUMAN ETHICS COMMITTEE**



**LOW RISK APPLICATION FORM**

(For research proposals which are not considered in full by the University Human Ethics Committee)

*FOR STUDENT RESEARCH UP TO AND INCLUDING MASTERS LEVEL*

**ETHICAL APPROVAL OF LOW RISK RESEARCH INVOLVING  
HUMAN PARTICIPANTS REVIEWED BY DEPARTMENTS**

*Please read the important notes appended to this form before completing the sections below*

- 1 RESEARCHER'S NAME: Jin Kwang Hong
- 2 NAME OF DEPARTMENT OR SCHOOL: Department of Computer Science and Software Engineering
- 3 EMAIL ADDRESS: jkh601@uclive.ac.nz
- 4 TITLE OF PROJECT: Identifying and responding to disengagement in a constraint-based intelligent tutoring system.
- 5 PROJECTED START DATE OF PROJECT: 27/02/2014
- 6 STAFF MEMBER/SUPERVISOR RESPONSIBLE FOR PROJECT: Prof. Tanja Mitrovic
- 7 NAMES OF OTHER PARTICIPATING STAFF AND STUDENTS: Dr. Kourosh Neshatian
- 8 STATUS OF RESEARCH: (e.g. class project, thesis) COSC460 Research Project

9 BRIEF DESCRIPTION OF THE PROJECT:

Please give a brief summary (approx. 300 words) of the nature of the proposal in lay language, including the aims/objectives/hypotheses of the project, rationale, participant description, and procedures/methods of the project:

This project focuses on predicting students' abandoning of problems given by SQL-Tutor, an intelligent tutoring system developed by the Intelligent Computer Tutoring Group (ICTG) from University of Canterbury. A predictor has been learned from past data of students using SQL-Tutor and this project's goal is to find out whether the predictor is accurate on new students at identifying whether students will abandon or not. Once a prediction has been made, if the prediction is that the student will abandon, an intervention message will pop up showing a motivational message to the student to see whether a motivational message is effective at keeping students engaged in the system. The participants will be University of Canterbury students taking the COSC265 course. The experiment will be held during one of the assigned lab sessions in term 4. The lab sessions are two hours in length.

10 WHY IS THIS A LOW RISK APPLICATION?

Description should include issues raised in the Low Risk Checklist.

Please give details of any ethical issues which were identified during the consideration of the proposal and the way in which these issues were dealt with or resolved.

This is a low risk application because it does not raise any issue of deception, threat, invasion of privacy, mental, physical or cultural risk or stress, and does not involve keeping personal information of sensitive nature about individuals. All participants will be given an information sheet describing the research project and will give consent for their results to be used for analysis. The participants will be fully aware that they can withdraw themselves or their results from the research at any stage. All the data will be kept in a secure environment and only those listed above allowed access to it.

**11 PROVIDE COPIES OF INFORMATION & CONSENT FORMS FOR PARTICIPANTS**

These forms should be on University of Canterbury Departmental letterhead. The name of the project, name(s) of researcher(s), contact details of researchers and the supervisor, names of who has access to the data, the length of time the data is to be stored, that participants have the right to withdraw participation and data provided without penalty, and what the data will be used for should all be clearly stated. A statement that the project has been reviewed approved by the appropriate department and the University of Canterbury Human Ethics Committee Low Risk Approval process should also be included.

*Please ensure that Section A, B and C below are all completed*

APPLICANT'S SIGNATURE: ..... Date .....

A SUPERVISOR DECLARATION:

- 1 I have made the applicant fully aware of the need for and the requirement of seeking HEC approval for research involving human participants.
- 2 I have ensured the applicant is conversant with the procedures involved in making such an application.
- 3 In addition to this form the applicant has individually filled in the full application form which has been reviewed by me.

SIGNED (Supervisor): ..... Date .....

B SUPPORTED BY THE DEPARTMENTAL/SCHOOL RESEARCH COMMITTEE:

Name .....

Signature: ..... Date .....

C APPROVED BY HEAD OF DEPARTMENT/SCHOOL:

Name .....

Signature: ..... Date .....

SUBMISSION OF APPLICATION:

- Please attach copies of any Information Sheet and Consent Form.
- Forward two hard copies to: The Secretary, Human Ethics Committee, Okeover House.

NOTES ON PROCEDURE:

- The Chair of the University of Canterbury Human Ethics Committee will review this application.
- In normal circumstances queries will be forwarded via email to the applicant within 7 days.
- Please include a copy of this form as an appendix in your thesis or course work.

---

ACTION TAKEN BY HUMAN ETHICS COMMITTEE:

Added to Low Risk Reporting Database

Referred to University of Canterbury HEC

Referred to another Ethics Committee - please specify:

.....

Reviewed by: .....

Date .....

## NOTES CONCERNING LOW RISK REPORTING SHEETS

1. This form should only be used for proposals which are Low Risk as defined in the University of Canterbury Human Ethics Committee Principles and Guidelines policy document, and which may therefore be properly considered and approved at departmental level under Section 5 of that document.
2. Low Risk applications are:
  - a Masters theses where the projects do not raise any issue of deception, threat, invasion of privacy, mental, physical or cultural risk or stress, and do not involve gathering personal information of a sensitive nature about or from individuals.
  - b Masters level supervised projects undertaken as part of specific course requirements where the projects do not raise any issue of deception, threat, invasion of privacy, mental, physical or cultural risk or stress, and do not involve gathering personal information of sensitive nature about or from individuals.
  - c Undergraduate and Honours class research projects which do not raise any issue of deception, threat, invasion of privacy, mental, physical or cultural risk or stress, and do not involve gathering personal information of sensitive nature about or from individuals, but do not have blanket approval as specified in Section 4 of the Principles and Guidelines.
3. No research can be counted as low risk if it involves:
  - (i) invasive physical procedures or potential for physical harm
  - (ii) procedures which might cause mental/emotional stress or distress, moral or cultural offence
  - (iii) personal or sensitive issues
  - (iv) vulnerable groups
  - (v) Tangata Whenua
  - (vi) cross cultural research
  - (vii) investigation of illegal behaviour(s)
  - (viii) invasion of privacy
  - (ix) collection of information that might be disadvantageous to the participant
  - (x) use of information already collected that is not in the public arena which might be disadvantageous to the participant
  - (xi) use of information already collected which was collected under agreement of confidentiality
  - (xii) participants who are unable to give informed consent
  - (xiii) conflict of interest e.g. the researcher is also the lecturer, teacher, treatment-provider, colleague or employer of the research participants, or there is any other power relationship between the researcher and the research participants.
  - (xiv) deception
  - (xv) audio or visual recording without consent
  - (xvi) withholding benefits from "control" groups
  - (xvii) inducements
  - (xviii) risks to the researcher

*This list is not definitive but is intended to sensitise the researcher to the types of issues to be considered. Low risk research would involve the same risk as might be encountered in normal daily life.*

### 4. Responsibility

*Supervisors are responsible for:*

- (i) Theses where the projects do not raise any issues listed below.
- (ii) Masters level supervised projects undertaken as part of specific course requirements where the projects do not raise any issue.
- (iii) Undergraduate and Honours class research projects which do not raise any issue listed but do not have blanket approval as specified in the Principles and Guidelines.

*Heads of Department are responsible for:*

- (i) Giving final departmental approval for the low risk application.
  - (ii) Ensuring a copy of all applications are kept on file in the Department/School.
5. A separate low risk form should be completed for each teaching or research proposal which involves human participants and for which ethical approval has been considered or given at Departmental level.
  6. The completed and signed Application form together with copies of any Information Sheet or Consent Form should be submitted to the Secretary, Human Ethics Committee, Okeover, as soon as the proposal has been considered at departmental level.
  7. The Information Sheet and Consent Form should include the statement “This proposal has been reviewed and approved by the Department of ....., University of Canterbury and the University of Canterbury Human Ethics Committee Low Risk process.”
  8. Please ensure the Consent Form and the Information Sheet have been carefully proof-read; the institution as a whole is likely to be judged by them.
  9. The research must be consistent with the University of Canterbury Human Ethics Committee Principles and Guidelines. Refer to the appendices of the UC HEC Principles and Guidelines for guidance on information sheets and consent forms.
  10. Please note that if the nature, procedures, location or personnel of the research project changes after departmental approval has been given in such a way that the research no longer meets the conditions laid out in Section 5 of the Principles and Guidelines, a full application to the Human Ethics Committee must be submitted.
  11. This form is available electronically at: <http://www.canterbury.ac.nz/humanethics>

## **CHECKLIST**

*Please check that your application/summary has discussed:*

- Procedures for voluntary, informed consent
- Privacy & confidentiality
- Risk to participants
- Obligations under the Treaty of Waitangi
- Needs of dependent persons
- Conflict of interest
- Permission for access to participants from other individuals or bodies
- Inducements

In some circumstances research which appears to meet low risk criteria may need to be reviewed by the University of Canterbury Human Ethics Committee. This might be because of requirements of:

- The publisher of the research.
- An organisation which is providing funding resources, existing data, access to participants etc.
- Research which meets the criteria for review by a Health and Disability Ethics Committee – see HRC web site.

If you require advice on the appropriateness of research for low risk review, please contact the Chair of the University of Canterbury Human Ethics Committee.

Department of Computer Science and Software  
Engineering  
Telephone: +64 322 305 7298  
Email: [jkh601@uclive.ac.nz](mailto:jkh601@uclive.ac.nz)  
Date: 03-09-2014



## **Identifying and responding to disengagement in a constraint-based intelligent tutoring system**

### **Consent Form for Study Participants**

I have been given a full explanation of this project and have had the opportunity to ask questions.

I understand what is required of me if I agree to take part in the research.

I understand that participation is voluntary and I may withdraw at any time without penalty. Withdrawal of participation will also include the withdrawal of any information I have provided should this remain practically achievable.

I understand that any information or opinions I provide will be kept confidential to the researcher and that any published or reported results will not identify the participants.

I understand that all data collected for the study will be kept in locked and secure facilities and/or in password protected electronic form and will be kept indefinitely in secure electronic form.

I understand the risks associated with taking part and how they will be managed.

I understand that I am able to receive a report on the findings of the study by contacting the researcher at the conclusion of the project.

I understand that I can contact the researcher Jin Kwang Hong ([jkh601@uclive.ac.nz](mailto:jkh601@uclive.ac.nz)) or supervisors Prof Tanja Mitrovic ([tanja.mitrovic@canterbury.ac.nz](mailto:tanja.mitrovic@canterbury.ac.nz)) and Dr Kourosh Neshatian ([kourosh.neshatian@canterbury.ac.nz](mailto:kourosh.neshatian@canterbury.ac.nz)) for further information. If I have any complaints, I can contact the Chair of the University of Canterbury Human Ethics Committee, Private Bag 4800, Christchurch ([human-ethics@canterbury.ac.nz](mailto:human-ethics@canterbury.ac.nz))

By signing below, I agree to participate in this research project.

Name: \_\_\_\_\_

Signed: \_\_\_\_\_

Date: \_\_\_\_\_

Please return this form to Jin Kwang Hong upon undertaking the study.

-Jin



HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen  
Email: [human-ethics@canterbury.ac.nz](mailto:human-ethics@canterbury.ac.nz)

Ref: HEC 2014/62/LR

17 September 2014

Jin Hong  
Department of Computer Science and Software Engineering  
UNIVERSITY OF CANTERBURY

Dear Jin

Thank you for forwarding your Human Ethics Committee Low Risk application for your research proposal "Identifying and responding to disengagement in a constraint-based intelligent tutoring system".

I am pleased to advise that this application has been reviewed and I confirm support of the Department's approval for this project.

Please note that this approval is subject to the incorporation of the amendments you have provided in your email of 12 September 2014.

With best wishes for your project.

Yours sincerely



Lindsey MacDonald  
***Chair, Human Ethics Committee***